Архитектура I

Кристиян Стоименов

1 ноември 2023 г.

ТУЕС,
ПВМКС

Ядро

Какво е архитектура?

Какво е архитектура?

Зависи :/

Какво е архитектура?

Зависи :/

- ISA (*instruction set architecture*);
- Микроархитектура
- Хардуер

Какво е архитектура?

Зависи :/

- ISA (*instruction set architecture*);
- Микроархитектура
- Хардуер

Обикновено някое от тях.

ISA

- Предназначние/клас (*class of ISA*) - *general purpose* най-вероятно; *register-memory & load-store*;

ISA

- Предназначние/клас (*class of ISA*) - *general purpose* най-вероятно; *register-memory* & *load-store*;
- Достъп до паметта - дали се допускат *unaligned* достъпи;

ISA

- Предназначние/клас (*class of ISA*) - *general purpose* най-вероятно; *register-memory* & *load-store*;
- Достъп до паметта - дали се допускат *unaligned* достъпи;
- Режими на адресиране;

ISA

- Предназначние/клас (*class of ISA*) - *general purpose* най-вероятно; *register-memory* & *load-store*;
- Достъп до паметта - дали се допускат *unaligned* достъпи;
- Режими на адресиране;
- Операции и операнди;

ISA

- Предназначние/клас (*class of ISA*) - *general purpose* най-вероятно; *register-memory* & *load-store*;
- Достъп до паметта - дали се допускат *unaligned* достъпи;
- Режими на адресиране;
- Операции и операнди;
- и т.н.

Микроархитектура

Обаче, `AMD Operton` и `Intel Core i7` имплементират едно и също ISA (80x86), а са различни архитектури.

Микроархитектура

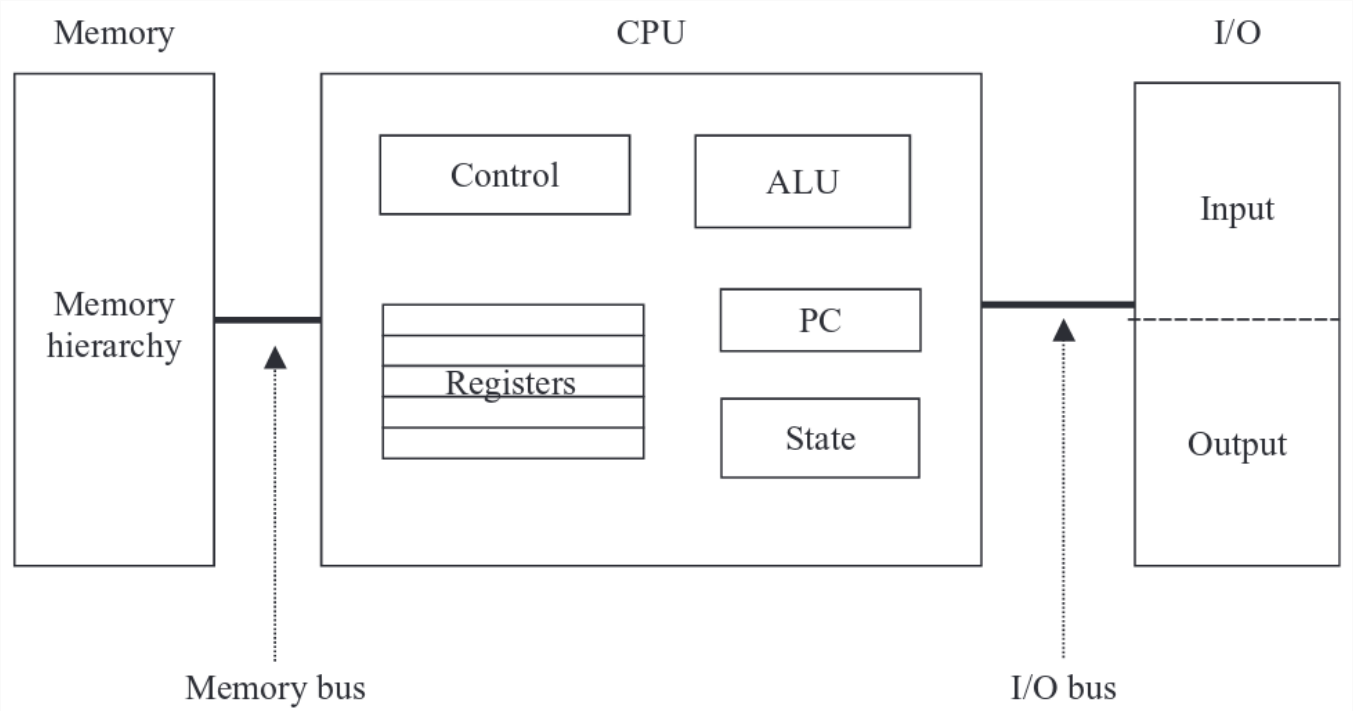
Обаче, `AMD Operton` и `Intel Core i7` имплементират едно и също ISA (80x86), а са различни архитектури. Това е поради разлики в микроархитектурата - т.е начина, по който е имплементиран ISA. Някои по-интересни аспекти (сред много други) са например

Микроархитектура

Обаче, `AMD Operton` и `Intel Core i7` имплементират едно и също ISA (80x86), а са различни архитектури. Това е поради разлики в микроархитектурата - т.е начина, по който е имплементиран ISA. Някои по-интересни аспекти (сред много други) са например

- Кешове;
- Out-of-order изпълнение;
- Преименуване на регистри.

Общ модел i


Memory
CPU
I/O
Memory hierarchy
Control
ALU
Registers
PC
State
Input
Output
Memory bus
I/O bus

Общ модел

Фундаменталните компоненти са:

- Процесор

Общ модел

Фундаменталните компоненти са:

- Процесор, съдържащ АЛУ, регистри, контролен блок и т.н;

Общ модел

Фундаменталните компоненти са:

- Процесор, съдържащ АЛУ, регистри, контролен блок и т.н;
- Памет

Общ модел

Фундаменталните компоненти са:

- Процесор, съдържащ АЛУ, регистри, контролен блок и т.н;
- Памет, съхраняваща самите програми и данни, необходими им;

Общ модел

Фундаменталните компоненти са:

- Процесор, съдържащ АЛУ, регистри, контролен блок и т.н;
- Памет, съхраняваща самите програми и данни, необходими им;
- Вход/изход

Общ модел

Фундаменталните компоненти са:

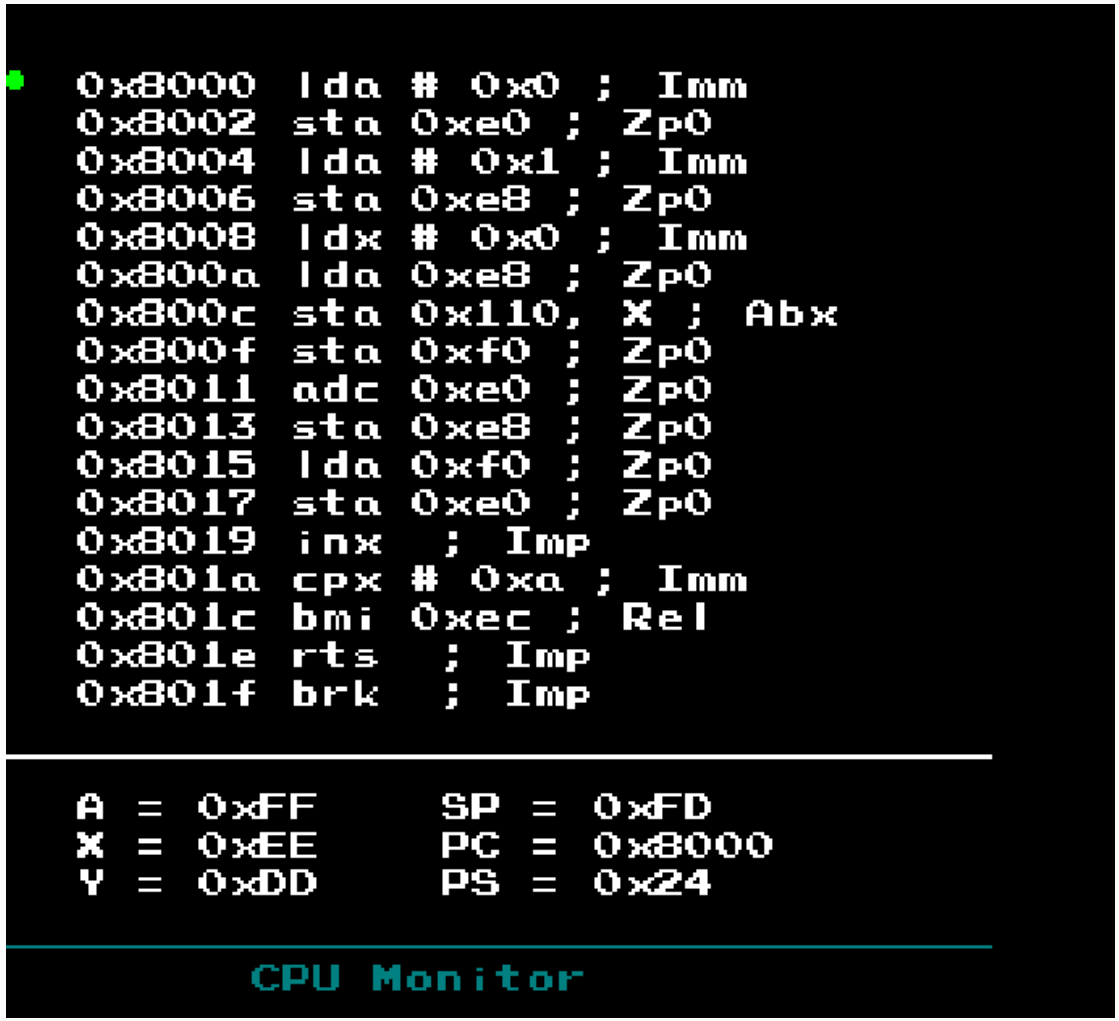
- Процесор, съдържащ АЛУ, регистри, контролен блок и т.н;
- Памет, съхраняваща самите програми и данни, необходими им;
- Вход/изход, служещи за комуникация с външния свят.

Изпълнение на една инструкция [1]:

1. Извличане (*fetch*);
2. Декодиране (*decode*);
3. Изпълнение (*execute*);

[1] Това описани е доста общо и до голяма степен неточно. За повече информация виж *pipelining*.

Пример с емулатор



Chapter 6. AVR CPU Core

Прекъсвания i

- Комуникационен механизъм, алтернатива на *polling*;
- Дейността се поддържа от `(A)PIC`;
- Bird's eye view - процесорът паузира и изпълнява нещо кратичко;
- *Level-triggered* и *edge-triggered*;
- Избира се най-високо приоритетното прекъсване;
- Започва изпълнение на *interrupt handler* на следващата инструкция.

Прекъсвания ii

Някои основни видове прекъсвания (при `Atmel 328P`):

- Външни (*external*);
- При промяна на състоянието (*pin change*);
- Свързани с таймерите (*timer overflow/compare match*);
- Свързани с интерфейсите (*SPI transfer complete; USART complete*);
- Други (*ADC complete; EE Ready*).

Как “възникват” прекъсванията?

- Общ процес е описан на стр. 15.
- Какво е *interrupt vector*? стр. 50

Chapter 11. Interrupts

Chapter 11. Interrupts
&
Chapter 12. External Interrupts

Как използваме външните прекъсвания?

- Четем Register Description, стр. 54

Как използваме външните прекъсвания?

- Четем Register Description, стр. 54
- `EICRA`, `EIMSK` & `EIFR`;

Как използваме външните прекъсвания?

- Четем Register Description, стр. 54
- `EICRA`, `EIMSK` & `EIFR`;

Пример

Да закачим бутон с прекъсване, конфигурирано чрез регистри, който да променя състоянието на вградения светодиод.

 решение

Как работи `attachInterrupt()`?

Как работи `attachInterrupt()`?

```
$ git clone
git@github.com:arduino/ArduinoCore-avr.git
$ git grep attachInterrupt
```

Как работи `attachInterrupt()`?

```
$ git clone
git@github.com:arduino/ArduinoCore-avr.git
$ git grep attachInterrupt
```

Бонус задача.

Как използваме прекъсвания при промяна в състоянието?

- Четем Register Description, стр. 56

Как използваме прекъсвания при промяна в състоянието?

- Четем Register Description, стр. 56
- `PCICR`, `PCIFR` & `PSMSKx`;

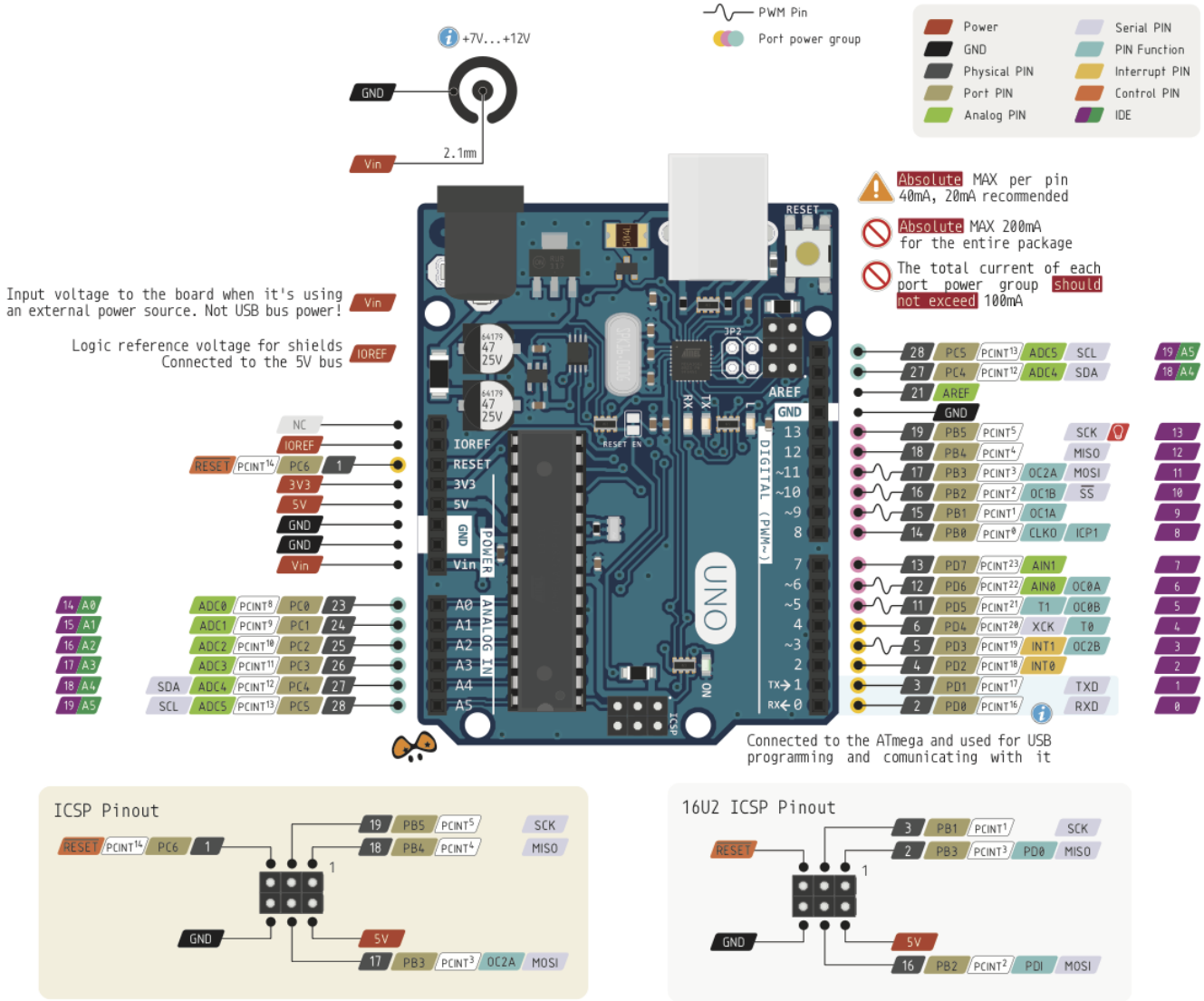
Как използваме прекъсвания при промяна в състоянието?

- Четем Register Description, стр. 56
- `PCICR`, `PCIFR` & `PSMSKx`;

Пример

Да закачим бутон с прекъсване, конфигурирано чрез регистри, който да променя състоянието на вградения светодиод.

решение


PWM Pin
Port power group
Power
GND
Physical PIN
Port PIN
Analog PIN
Serial PIN
PIN Function
Interrupt PIN
Control PIN
IDE
+7V...+12V
GND
Vin
2.1mm
Absolute MAX per pin 40mA, 20mA recommended
Absolute MAX 200mA for the entire package
The total current of each port power group should not exceed 100mA
Input voltage to the board when it's using an external power source. Not USB bus power!
Logic reference voltage for shields Connected to the 5V bus
Connected to the ATmega and used for USB programming and comunicating with it
ICSP Pinout
16U2 ICSP Pinout

Литература

- ***"Atmel328P Datasheets"***. URL: https://gitlab.com/tues-embedded/vmks/-/blob/master/DatasheetsAtmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf?ref_type=heads/ (дата на посещ. 28.09.2023)
- Jean Loup-Baer. ***"Microprocessor architecture"***. 2010
- "Joch L. Henessy и David A. Peterson". ***"Computer Architecture"***. 2019